Inductive versus approximative learning

Wolfram Menzel and Frank Stephan

1 Introduction

A central and crucial point in any mathematical model of learning is the notion of "convergence" used to express the success of the process being studied. In the wide variety of approaches to learning, mainly two types of convergence concepts appear to be used, we call them *approximative* and *inductive*, respectively.

Let the usual scenario of learning functions from pairs 'argument, function value' be given. *Approximative* learning means that when more and more data are provided, the corresponding hypotheses will converge in probability to the target function (or a best possible approximation to it): achieving it only in the limit and only in a probabilistic sense, but, on the other hand, in such a way that the quality of the hypotheses at given times can in fact be estimated. In contrast to this, successful *inductive* learning means that the generating of hypotheses will at some time "click into its right place", i.e., become absolutely correct and stay so for ever. But this is not for free: one pays the price that, in general, the learner will never know how good its current hypothesis really is. Approximation is convergence with respect to a specific distance measure between functions, while inductive convergence is convergence w.r.t. the discrete distance measure and hence to any distance measure whatever. Inductive convergence means that in the "discrete" space of a *language*, designed to represent functions, an expression (program) is eventually found which in fact identifies the searched function, while approximative convergence corresponds to a more semantic-oriented search in the given function space.

It appears to be a natural desire to combine the benefits of both convergence concepts, i.e., to combine the ability to ultimately find a definitely correct program after finitely many observations with the possibility to estimate the quality of any current hypothesis. This is the objective of our present contribution. We aim at bringing the view of statistical learning theory into the world of computable functions, say (in the usual standardization) from $\mathbb{N}$ to $\mathbb{N}$, and investigate to what extent inductive learning can be accompanied with a gradual and measurable improvement of the hypotheses. Our results will rest heavily upon the simple fact that, in contrast to more general situations, any probability distribution over a countable domain has the property that there are always finite sets with arbitrarily high probabilities less than 1.

$\mathbb{N}$ is the set of natural numbers including 0. We assume familiarity with the basic notions from computability theory and probability measures. REC is the class of total recursive, shortly *recursive* functions from $\mathbb{N}$ to $\mathbb{N}$, PREC the corresponding class of *partial recursive* functions. Small Latin letters $f, g, \ldots$ will be used to denote total functions, small Greek letters $\phi, \psi, \ldots$ for denoting partial ones. $\psi(x) \downarrow$ means that ψ is defined at argument x, $\psi(x) \uparrow$ that it is not. $dom(\psi) = \{x : \psi(x) \downarrow\}$. At some places, computable functions with arguments and/or function values other than natural numbers will be used in the familiar way. A *numbering* is a two-place partial recursive function ψ, and for any $e \in \mathbb{N}$, ψ_e is the function in PREC with $\psi_e(x) = \psi(e, x)$. ψ *enumerates* the class $\{\psi_e : e \in \mathbb{N}\} \subseteq$ PREC, and any $F \subseteq$ PREC is *recursively enumerable* (*r.e.*) if it can be enumerated by an appropriate two-place partial recursive function ψ. PREC itself is r.e., REC is not. Among the numberings enumerating the whole of PREC there are particular ones, called *acceptable*, which correspond to the functional semantics of universal programming languages (where legal programs are identified with their respective numbers in some effective listing). There are several equivalent characterizations of being acceptable, e.g., ψ is acceptable iff any numbering ϱ can be translated into ψ in a computable way. As usual, we will distinguish, and henceforth keep fixed, some particular acceptable numbering φ as our "standard" basis for computing. A class $S \subseteq$ PREC is *dense* iff every finite partial function σ can be extended by some total $f \in S$.

We will deal with probability distributions on $\mathbb{N}$, hence the corresponding probability measures are always on the whole power set $\{0, 1\}^{\mathbb{N}}$. Derived probabilities over finite products of $\mathbb{N}$, or infinite products such as $\mathbb{N}^\infty$, $(\mathbb{N} \times \mathbb{N})^\infty$, will then occur in the usual way.

The two roots of our considerations, viz., inductive inference and probabilistic approximation, are presented in Sections 2 and 3, respectively. Section 4 is devoted to combining both approaches in a distribution-dependent way, and Section 5 to the corresponding combining in a distribution-free way.

2 Inductive Learning

Inductive inference was introduced by Gold [3]. It conceives learning as finding from finitely many examples a *rule* or *program* to compute an initially unknown function. As a consequence, the functions to be learned are computable, and we restrict ourselves to the standard setting of functions from $\mathbb{N}$ to $\mathbb{N}$. As pointed out in the introduction, a *program* or *index* for a function f is an $e \in \mathbb{N}$ with $\varphi_e = f$. A *learner* is a machine which reads the graph of an unknown f in growing, finite portions and from time to time, during this process, puts out a number as a *hypothesis* for a program of f.

Definition 2.1 (Learner). A **learner** is a program for a computable partial function

$$L : (\mathbb{N} \times \mathbb{N})^* \to \mathbb{N} \cup \{?\},$$

where *?* is a special extra symbol, and for all sequences $((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathbb{N} \times \mathbb{N})^*$, if $L((x_1, y_1), \ldots, (x_m, y_m)) \downarrow$ then $L((x_1, y_1), \ldots, (x_k, y_k)) \downarrow$ for all $k \leq m$.

$L((x_1, y_1), \ldots, (x_m, y_m)) \downarrow= ?$ is an abbreviation for $L((x_1, y_1), \ldots, (x_m, y_m)) \downarrow$ and $L((x_1, y_1), \ldots, (x_m, y_m)) = ?$. Correspondingly, $L((x_1, y_1), \ldots, (x_m, y_m)) \downarrow= e$ for an $e \in \mathbb{N}$. Interpret $L((x_1, y_1), \ldots, (x_m, y_m)) \downarrow= e$ as: The learner having read examples $(x_1, y_1), \ldots, (x_m, y_m)$ outputs e as a *hypothesis* for a program of the unknown function. *The term "hypothesis" will never be used for the symbol ?*. We shortly write $L((x_1, y_1), \ldots, (x_m, y_m)) \downarrow \in \mathbb{N}$ for: There is an $e \in \mathbb{N}$ with $L((x_1, y_1), \ldots, (x_m, y_m)) \downarrow = e$. Read $L((x_1, y_1), \ldots, (x_m, y_m)) \downarrow =$ *?* as: The learner has decided not to output a hypothesis but instead to request to see the next data pair. $L((x_1, y_1), \ldots, (x_m, y_m)) \uparrow$ means that the learner has got stuck in never ending internal computations. For $m = 0$, $((x_1, y_1), \ldots, (x_m, y_m))$ is the empty sequence λ. Note that we did not stipulate anything specific for ambiguous sequences, where $x_i = x_j, y_i \neq y_j$ for some i, j. The learner might have a strategy for getting along with "disturbances" or the like, but ambiguous sequences will not have any importance for our present considerations.

The learner is *total* iff $L((x_1, y_1), \ldots, (x_m, y_m)) \downarrow$ for all sequences $((x_1, y_1), \ldots, (x_m, y_m))$. *For the sake of simplicity, also the function L itself* (*instead of a program for L*) *will be called "learner"*.

Thus, given an infinite sequence $(x_1, f(x_1)), (x_2, f(x_2)), \ldots$ coming from some f, a learner produces a finite or infinite sequence of hypotheses $e_0, e_1, \ldots$, sometimes remaining tacit for a while (i.e., producing *?*) because of the recognized need for more information, and possibly remaining tacit for ever in a non-recognized manner because of never ending computing. In general, a learner is constructed with respect to a class S of functions: Information about S may be built into L, while the "learning" consists of additionally identifying a particular $f \in S$. Definition 2.2 below expresses learning in the mode of *syntactical convergence*. Here, an infinite sequence $(x_1, f(x_1)), (x_2, f(x_2)), \ldots$ is called a *valid description* of f if every $x \in \mathbb{N}$ occurs among the x_k.

Definition 2.2 (Ex-learning[3]). A class $S \subseteq$ REC is **Ex-learnable** (**explanatorily learnable**) iff there is a learner L such that for all $f \in S$ and all valid descriptions $(x_1, f(x_1)), (x_2, f(x_2)), \ldots$ of f,

- there are infinitely many numbers m with $L((x_1, f(x_1)), \ldots, (x_m, f(x_m))) \downarrow \in \mathbb{N}$;
- and there is an $e \in \mathbb{N}$ such that $\varphi_e = f$ and $L((x_1, f(x_1)), \ldots, (x_m, f(x_m))) \downarrow \in \{e, ?\}$ for almost all m.

Note that we have *not* demanded that from some "convergence point" onward no more *?* will occur. This is because of our intended combining with approximation, but it fits perfectly well to the idea of inductive inference: As the learner never knows about its possibly reached success, "thinking phases" may well occur in that "stable" period, too. (Anyway it is always possible to dispense with *?* at all by defining $L(\lambda) = 0$ and afterwards simply repeating the last hypothesis until the next one appears; but this amounts to a loss of information unfavourable for our further analysis.)

As has been remarked in the Introduction, REC is not recursively enumerable. But there are rather rich subclasses S of REC which are (e.g., the primitive recursive functions), i.e., there is a total recursive g with $S = \{g_0, g_1, \ldots\}$. Every such class is Ex-learnable:

Example 2.3 (Learning by Enumeration). Given an r.e. class S of total functions g_0, $g_1, g_2, \ldots$, the algorithm **Learning by Enumeration** computes for any unambiguous input $((x_1, y_1), \ldots, (x_m, y_m))$ the first index e with $g_e(x_k) = y_k$ for $k = 0, \ldots, m$ and outputs $h(e)$, where h is a fixed computable translation from g-indices to φ-indices.

Evidently, this algorithm learns any g_i from any valid description of g_i. Note that the corresponding L might be partial, but can always be made total: Enrich S by adding some dense r.e. subclass of REC (e.g., the almost-everywhere-0 functions), then the resulting algorithm **Learning by Enumeration** for the new class will be a total Ex-learner for the new class, a fortiori for S.

There are Ex-learnable classes of total recursive functions which are not contained in any r.e. subclass of REC. A prominent example is the class of *self-describing* functions $S = \{f : f \in \text{REC}, f = \varphi_{f(0)}\}$. A learner for S simply waits (produces *?*) until some example $(0, e)$ shows up and from then onward outputs e. But S is not contained in any r.e. subclass of REC: For any total numbering g there is a self-describing f not in $\{g_0, g_1, \ldots\}$. Define functions f_i by $f_i(0) = i$ and $f_i(x+1) = g_x(x+1) + 1$. By the recursion theorem [7,9], there is an e with $f_e = \varphi_e$. f_e is self-describing and not contained in $\{g_0, g_1, \ldots\}$.

Bārzdiņš [1] generalized Ex-learning to the wider concept of learning in the sense of *semantical convergence*, called *BC-learning* ("behaviourally correct"). L BC-learns f iff any valid description of f causes L to produce infinitely many hypotheses, almost all of which are indices of f. Thus in contrast to Ex-learning, the set of output hypotheses $\{e_0, e_1, \ldots\}$ may be infinite.

Definition 2.4 (BC-learning [1]). $S \subseteq$ REC is **BC-learnable** (**behaviourally correct**) iff there is a learner L such that for all $f \in S$ and all valid descriptions $(x_1, f(x_1))$, $(x_2, f(x_2)), \ldots$ of f,

- there are infinitely many m with $L((x_1, f(x_1)), \ldots, (x_m, f(x_m))) \downarrow \in \mathbb{N}$; and
- for almost all output hypotheses e, $\varphi_e = f$.

Clearly, every Ex-learner of some S is also a BC-learner of S. But there are BC-learnable classes which are not Ex-learnable. An example is

$$\{f : f \in \text{ REC}, \varphi_{f(0)} \text{ is a finite variant of } f\}.$$

The BC-learner for this class waits until an example $(0, e)$ has appeared and from then on outputs an index for that finite variant of φ_e which is obtained by patching all observed values into φ_e:

$$\varphi_{L((x_1,y_1),\ldots,(x_m,y_m))}(x) = \begin{cases} y_k & \text{if } x = x_k \text{ for some } k \in \{1, \ldots, m\} \\ \varphi_e(x) & \text{otherwise.} \end{cases}$$

After having inspected enough data, all undefined places will have been patched and from then onwards, L's hypotheses are indices of total functions. As it is the undefined places which cause a problem (not those carrying false values) and it is not possible to detect them in the limit, one can show that it is not possible to Ex-learn the above class [2].

3 Probabilities and Approximation

We will now deal with the statistical approach to learning, which is the second root of our questions. As we are concentrating upon computable functions from $\mathbb{N}$ to $\mathbb{N}$, probability distributions over $\mathbb{N}$ are the natural basis for further analysis. It is common use in this context to apply such a probability distribution pr not only for describing the occurrence of data, but also for establishing a corresponding distance measure between functions (see Vidyasagar [12] for a discussion of this point). Thus for total functions f, g we define

$$D_{pr}(f, g) = pr(\{x : f(x) \neq g(x)\}).$$

Similarly for partial functions ϕ, ψ:

$$D_{pr}(\phi, \psi) = pr(\{x : \phi(x) \uparrow \text{ or } \psi(x) \uparrow \text{ or } \phi(x) \downarrow \neq \psi(x) \downarrow\}),$$

so that functions are considered to disagree at any place where at least one of them is undefined (see [5] for a more detailed discussion of this view). Furthermore, we associate a probability distribution on $\mathbb{N} \times \mathbb{N}$ to pr and a total f by

$$pr_f(\{x, y\}) = \begin{cases} pr(\{x\}) & \text{if } y = f(x) \\ 0 & \text{otherwise .} \end{cases}$$

Note that, in contrast to "Lebesque-style" probability measures over subsets of $\mathbb{R}$ which are often considered in statistical learning theory (where countable sets have measure zero), pr over $\{0, 1\}^{\mathbb{N}}$ has the property that for all $\epsilon > 0$ there is a finite $E \subseteq \mathbb{N}$ with $pr(E) \geq 1 - \epsilon$. It will often be appropriate to demand a little more, namely $pr(\{x\}) > 0$ for all $x \in \mathbb{N}$, and probabilities with this property will be called *fair*. The main reason for focusing on fair probabilities is that these have the nice property that for infinite sequences $x_1, x_2, \ldots$,

$$pr^{\infty}(\{(x_1, x_2, \ldots) : (\forall x)(\exists k)[x = x_k]\}) = 1,$$

where pr^{∞} is the standard extension of pr to infinite sequences, using cylinder sets. Hence given f, an infinite sequence of examples for f will with probability 1 contain every $(x, f(x))$. As a consequence, $D_{pr}(\phi, \psi) = 0$ for partial functions ϕ, ψ if and only if ϕ, ψ are total and equal. Furthermore, on the total functions every metric D_{pr} generated by some fair pr is isometric to the *standard* metric [5,7] D_{sm} defined as

$$D_{sm}(f, g) = \begin{cases} 0 & \text{if } f = g \\ \frac{1}{min\{x : f(x) \neq g(x)\} + 1} & \text{otherwise.} \end{cases}$$

The isometry of D_{pr} and D_{sm} can be verified as follows. For all $\epsilon > 0$ there is $\delta > 0$ such that $D_{pr}(f, g) < \delta$ implies $D_{sm}(f, g) < \epsilon$, namely,

$$\delta = min\{pr(\{x\}) : \tfrac{1}{x+1} \geq \epsilon\}.$$

$\delta > 0$ because δ is the minimum of finitely many numbers greater than zero, and $D_{sm}(f, g) < \epsilon$ follows from the fact that $D_{pr}(f, g) < \delta$ implies $f(x) = g(x)$ for all x with $\frac{1}{x+1} \geq \epsilon$. As for the converse, given ϵ, define

$$x = min\{y : pr(\{0, \ldots, y\}) > 1 - \epsilon\},$$
$$\delta = \tfrac{1}{x+1}.$$

Then $D_{pr}(f, g) \leq 1 - pr(\{0, \ldots, x\}) < \epsilon$ whenever $D_{sm}(f, g) < \delta$. Note that these transformations are computable if ϵ and all $pr(\{x\})$ are rational numbers and the mapping $x \mapsto pr(\{x\})$ is computable.

Note that even if pr is fair, it is only with probability 1 that infinite sequences $(x_1, f(x_1)), (x_2, f(x_2)), \ldots$ generated according to pr_f are valid: Invalid sequences like $(0, f(0)), (0, f(0)), \ldots$ do exist, although altogether they have measure 0. Such invalid sequences might well contain some relevant information, as is the case for, e.g., $(0, f(0)), (0, f(0)), \ldots$ when the self-describing functions are to be learned. As a consequence, our notion of successful learning will merely mean success with probability 1.

Learning in statistical learning theory is uniform approximating in probability. It is defined in the PAC understanding ("probably approximately correct") as introduced by Valiant [10]. We will here follow the exposition by Vidyasagar [12], abstracting in our more principal questions from complexity issues, which are nonetheless crucially important for some kinds of applications.

Given pr and a function f to be learned, the learner sees data according to pr_f and produces *?* or hypotheses. Now observe that if for a current hypothesis e one can decide $dom(\varphi_e)$ then it is possible to check algorithmically whether φ_e is near enough to f. More generally, for any decidable $A \subseteq \mathbb{N}$ let the *test for* A be the random variable X_A defined by evaluating the characteristic function of A on some drawn x, and the *success probability* the corresponding probability that $x \in A$. Call any individual drawing of an x a *trial*. Then the *Chernoff bound* (see, e.g., [12]) allows us to estimate the success probability p of X_A. Namely, given $\epsilon > 0$ (w.l.o.g. a rational number), if trials on A are repeated k times and in h cases succeed then the probability that $|p - \frac{h}{k}| < \epsilon$ is at least $1 - 2 \cdot exp(-2k\epsilon^2)$. We have the following

Fact 3.1 (Quality of Tests). *Let pr be given, and let X be the test for any (in our cases: recursive) property of natural numbers. Then one can for any $\epsilon, \delta > 0$ (w.l.o.g. rational numbers) compute a number $cher(\epsilon, \delta) = \ulcorner \frac{1}{2\epsilon^2}\, ln\, (\frac{1}{\delta}) \urcorner$ of trials such that the success probability p of X can be estimated as follows. If h among $cher(\epsilon, \delta)$ trials on X are successful then it holds with probability at least $1 - \delta$ that $|p - \frac{h}{cher(\delta, \epsilon)}| < \epsilon$.*

$1 - \delta$ above is commonly called the *confidence level* achievable for given X and ϵ. Our main application of Fact 3.1 will be that a learner can delay its next hypothesis (i.e., produce *?*) until a certain consistency test has been passed with high probability, this way ensuring sufficiently good hypotheses throughout.

In the context of learning, the above "trials" will be called *drawn examples* (from $\mathbb{N}$ or $\mathbb{N} \times \mathbb{N}$); by the *number of examples* in a sequence $(x_1, \ldots, x_m)$ we mean the length m of the sequence and not the cardinality of the set $\{x_1, \ldots, x_m\}$.

Mostly for sake of convenience we will replace parameters ϵ, δ above by a single one, n with values in $\mathbb{N}$, which controls both the current error and adjoined confidence. Furthermore, n will also serve to count output hypotheses and thus provide a bridge to inductive behaviour.

For f and some $X \subseteq \mathbb{N}$, let $f \upharpoonright X$ be the restriction of f to X:

$$(f \upharpoonright X)(x) = \begin{cases} f(x) & \text{if } x \in X \\ \uparrow & \text{otherwise} \end{cases}$$

Much of our later results will rest on the following

Proposition 3.2 (The Basic Approximation Algorithm). *There is a never terminating probabilistic algorithm B such that the following holds for all probability distributions pr on $\mathbb{N}$ and all $n \in \mathbb{N}$.*

- *There exists a constant $s_{pr,n}$ such that B having seen $s_{pr,n}$ examples according to pr has with probability greater than $\frac{n}{n+1}$ gathered a finite set $X_{pr,n}$ satisfying $pr(X_{pr,n}) > \frac{n}{n+1}$, and has put it out.*

Hence $D_{pr}(f, f \restriction X_{pr,n}) < \frac{1}{n+1}$ with probability greater than $\frac{n}{n+1}$ for all total functions f (not necessarily computable). The sets $X_{pr,n}$ are generated for $n = 0, 1, 2, \ldots$ in order, and it is with probability 1 that they are all eventually generated.

Proof. B is the following algorithm.

1: $m = 0$, $n = 0$, $X = \emptyset$;
2: $i = 2$;
3: $k = 1 + cher(\frac{1}{4(n+1)}, \frac{2^{-i}}{n+1})$,
draw k examples $x_{m+1}, \ldots, x_{m+k}$ according to pr,
$h = |\{j : 1 \leq j \leq k,\ x_{m+j} \in X\}|$,
$X = X \cup \{x_{m+1}, \ldots, x_{m+k}\}$;
4: $m = m + k$, $i = i + 1$;
5: if $\frac{h}{k} \leq \frac{2n+1}{2n+2}$ then go to 3;
6: $X_{pr,n} = X$, output $X_{pr,n}$;
7: $n = n + 1$, go to 2.

The idea behind the computation of $X_{pr,n}$ is to postpone its output until enough data $x_1, \ldots, x_m$ have been seen so as to guarantee that $pr(\{x_1, \ldots, x_m\}) > \frac{n}{n+1}$. As pr is unknown, this cannot be checked directly, so the Chernoff bound is used to obtain an estimate for $pr(\{x_1, \ldots, x_m\})$ (steps 2 and 5).

In more detail, we conclude as follows.

(a) With probability 1, all $X_{pr,n}$ are eventually output:
We have to show that

$$pr^{\infty}(\ \{(x_1, x_2, \ldots) : (\forall n)[B \text{ on sequence } x_1, x_2, \ldots \text{ will at some time output } X_{pr,n}]\}) = 1,$$

where again pr^{∞} is the standard extension of pr to sets of infinite sequences. Let $p_0, p_1, \ldots$ be as follows: p_0 is the probability that $X_{pr,0}$ will be output, and p_n for $n \geq 1$ is the conditional probability that if $X_{pr,n-1}$ will be output then so will $X_{pr,n}$. When the algorithm has (in the case $n = 0$) finished step 1 or has (in the case $n \geq 1$) output $X_{pr,n-1}$ and then finished step 7, it enters the loop to compute $X_{pr,n}$. Then as long as $X_{pr,n}$ has not yet been output one is forced to draw more and more examples, so that from some point onward X will satisfy $pr(X) \geq \frac{4n+3}{4n+4}$. Henceforth for each

i, the probability that the algorithm has not terminated for i and corresponding values h, k, i.e., $\frac{h}{k} \leq \frac{2n+1}{2n+2}$ or equivalently

$$\frac{4n+3}{4n+4} - \frac{h}{k} \geq \frac{1}{4n+4},$$

is less or equal to the probability that $|pr(X) - \frac{h}{k}| \geq \frac{1}{4n+4}$, which is at most $\frac{2^{-i}}{n+1}$. It follows that the probability that the algorithm will never output $X_{pr,n}$ is zero. Thus, $p_n = 1$ for all n. But the set of infinite sequences causing $X_{pr,n}$ to be output is a subset of those ones causing $X_{pr,n-1}$ to be output. Hence (by induction on n) p_n is also the unconditional probability that $X_{pr,n}$ is eventually output, and

$$pr^{\infty}(\ \{(x_1, x_2, \ldots) : (\forall x)[B \text{ on sequence } x_1, x_2, \ldots \text{ will at some time output } X_{pr,n}]\}) = \inf_n p_n = 1.$$

(b) We have from (a) that for every n there is a first number a_n such that the probability that B outputs $X_{pr,n}$ for an $i \leq a_n$ is at least $\frac{2n+1}{2n+2}$. Define

$$s_{pr,n} = \sum_{n'=0}^{n} \sum_{i=2}^{a_{n'}} \left(1 + cher\left(\frac{1}{4(n+1)}, \frac{2^{-i}}{n+1}\right)\right).$$

(c) If $X_{pr,n}$ is output for a certain value of i then it holds with a probability of at most $\frac{2^{-i}}{n+1}$ that $|pr(X_{pr,n}) - \frac{2n+1}{2n+2}| \geq \frac{1}{4n+4}$, hence that $pr(X_{pr,n}) \leq \frac{n}{n+1}$. So the probability that $X_{pr,n}$ with $pr(X_{pr,n}) \leq \frac{n}{n+1}$ is output for some $i \leq a_n$, is at most $\sum_{i=2}^{a_n} \frac{2^{-i}}{n+1} < \frac{1}{2n+2}$. From the definition of a_n then, we have that the probability that up to $i = a_n$ either no output of $X_{pr,n}$ occurs or else the output of an $X_{pr,n}$ with $pr(X_{pr,n}) \leq \frac{n}{n+1}$, is less then $\frac{1}{2n+2} + \frac{1}{2n+2} = \frac{1}{n+1}$. The assertion now follows from the definition of $s_{pr,n}$.

(d) It is immediate from the definition of D_{pr} that $D_{pr}(f, f \upharpoonright X_{pr,n}) < \frac{1}{n+1}$ with probability greater than $\frac{n}{n+1}$ for all total functions f. $\square$

Let L be a learner. We regard its input sequences $((x_1, y_1), \ldots, (x_m, y_m))$ as read according to a distribution on $\mathbb{N} \times \mathbb{N}$. For L and a given infinite sequence $(x_1, f(x_1))$, $(x_2, f(x_2)), \ldots$, let e_n be the n'th corresponding output hypothesis in order, where outputs *?* are neglected. Because of technical reasons, we begin this counting at 0 (by Definition 2.1, e_0 may, but need not be $L(\lambda)$). If $x_1, x_2, \ldots$ are regarded as drawn according to some pr then e_n becomes a random variable w.r.t. pr, so that we may speak of the probability that e_n has some property we are interested in. From our Basic Approximation Algorithm then, we immediately obtain a learner with a rather far-reaching approximation capability.

Definition 3.3 (Approximate). Let S be a class of total functions. The learner L (uniformly) **approximates** S iff the following holds for all probability distributions pr on $\mathbb{N}$ and $n \in \mathbb{N}$. There is a constant $s_{pr,n}$ such that for all $f \in S$, when examples $(x_1, f(x_1)), (x_2, f(x_2)), \ldots$ are drawn w.r.t. pr_f and e_n is the n'th hypothesis of L w.r.t. the infinite sequence arising, then with probability greater than $\frac{n}{n+1}$, e_n will really be output and $D_{pr}(\varphi_{e_n}, f) < \frac{1}{n+1}$. L is a **universal approximator** iff L approximates the whole class of total functions from $\mathbb{N}$ to $\mathbb{N}$, including the non-computable ones.

Theorem 3.4 (Universal Approximation). *There exists a universal approximator.*

Proof. The learner U when reading examples according to pr_f just simulates B (on their x-parts) and, for $n = 0, 1, \ldots$, outputs its n'th hypothesis u_n if and when B outputs an $X_{pr,n}$ after the finite sequence seen so far. Thus, if $((x_1, y_1), \ldots, (x_m, y_m))$ is unambiguous and such that B after reading exactly $(x_1, \ldots, x_m)$ outputs $X_{pr,n}$, let u_n be such that

$$\varphi_{u_n}(x) = \begin{cases} y_k & \text{if } x = x_k \\ \uparrow & \text{otherwise.} \end{cases}$$

Then define $U((x_1, y_1), \ldots, (x_m, y_m))$ as u_n if $((x_1, y_1), \ldots, (x_m, y_m))$ is unambiguous, and B, if it has drawn the sequence $(x_1, \ldots, x_m)$ during its successive computation of $X_{pr,0}, X_{pr,1}, \ldots$, outputs $X_{pr,n}$ after having seen exactly $(x_1, \ldots, x_m)$. Otherwise, $U((x_1, y_1), \ldots, (x_m, y_m))$ is just *?*.

By Proposition 3.2, U is a universal approximator. $\square$

It is of course *not* generally the case that $\lim_{n\to\infty} \varphi_{e_n} = f$, in the sense of pointwise convergence, because pr might generate invalid sequences. (Pointwise convergence means that $\lim_{n\to\infty} \psi_n = f$ iff $(\forall x)(\exists n_x)(\forall n \geq n_x)[\psi_n(x) \downarrow = f(x)]$.) But remembering our remarks on fairness we have the following result.

Corollary 3.5 (Convergence to Target). *Let U be a universal approximator, and for any pr, f, n let $e_{pr,f,n}$ be the random variable, which is the n'th output hypothesis of U when examples are drawn according to pr_f (where $e_{pr,f,0}$ counts as the 0'th hypothesis). Then*

$$\lim_{n\to\infty} \varphi_{e_{pr,f,n}} = f \text{ with probability } 1$$

for all fair pr and all total $f : \mathbb{N} \to \mathbb{N}$, in the sense of pointwise convergence.

Remarks 3.6 (Dependency versus Knowing). The main point in Proposition 3.2 and Theorem 3.4 is of course that while $s_{pr,n}$ depends on pr, *the algorithms B and U themselves do not*. If we had allowed pr to be built in the algorithm then establishing a bound s such that after s examples a set X with $pr(X) > \frac{n}{n+1}$ has with confidence $\frac{n}{n+1}$ been collected would have been fully trivial. This is in contrast to the exposition in Vidyasagar [12], where the dependency of quality bounds from pr is somehow identified with the algorithm's "knowing" pr (pages 153, 195). In our opinion, such an understanding misses the central point.

A second remark is on Vidyasagar's distinguishing between PAC and PUAC learning, the latter being the stronger notion. In our context, L being PUAC on S for a given pr would mean: For every n, the probability of the set of sequences $(x_1, \ldots, x_m)$ satisfying

$$(\exists f \in S)\left[D_{pr}(\varphi_{L((x_1,f(x_1)),\ldots,(x_m,f(x_m)))}, f) \geq \frac{1}{n+1}\right]$$

goes to zero when m goes to infinity. It is easily verified that our universal approximator in fact has this property, w.r.t. the class of all total functions from $\mathbb{N}$ to $\mathbb{N}$ and any given pr.

4 Combining Inductive Learning with Approximation

We are now prepared to define our intended "combining" of inductive learning with approximation. In the following definition, we will speak of "types of inductive learning". Examples of such types are Ex, BC, and Num, the latter being learning by enumeration (Example 2.3), but there are a lot of others in the literature. The expression "L T-learns f with probability 1" below means that the set of sequences on which L T-learns f has the probability 1 with respect to pr_f.

Definition 4.1 (Combining Inductive Learning with Approximation). Let S be a class of total functions from $\mathbb{N}$ to $\mathbb{N}$, T a type of inductive learning, and L a learner.

L combines T-learning of S with approximation iff for all fair probabilities pr and $n \in \mathbb{N}$ there is a constant $s_{pr,n}$ such that for all $f \in S$,

- L T-learns f with probability 1;
- with probability greater than $\frac{n}{n+1}$: L has output its n'th hypothesis e_n after at most $s_{pr,n}$ examples read, and $D_{pr}(f, \varphi_{e_n}) < \frac{1}{n+1}$.

L combines T-learning of S **with universal approximation** iff the second requirement is satisfied for all total f (not only those in S, not only computable ones).

A first consequence of Theorem 3.4 refers to dense classes of functions. Recall that S is *dense* if every finite partial function possesses a total extension in S. Examples of dense classes are the total functions almost everywhere 0 and the primitive recursive functions.

Corollary 4.2 (Dense Classes). *Let S be a dense class of total functions, and let the learner L approximate S. Then L can be transformed into a universal approximator M by just skipping the first output hypothesis and afterwards behaving like L. Moreover, if L combines Ex-learning or BC-learning of S with approximation, respectively, then M does so with universal approximation.*

Proof. As indicated, $M((x_1, y_1), \ldots, (x_m, y_m)) = ?$ if $L((x_1, y_1), \ldots, (x_k, y_k)) \downarrow = ?$ for $k = 1, \ldots, m$, and $M((x_1, y_1), \ldots, (x_m, y_m)) = L((x_1, y_1), \ldots, (x_m, y_m))$ otherwise.

Let $s'_{pr,n}$ be the bound belonging to L according to Definition 3.3. We show that M is a universal approximator with bound $s_{pr,n} = s'_{pr,n+1}$. Define $r_n = \frac{1}{(n+1)(n+2)}$ and let A_n be a finite set with $pr(A_n)^{s_{pr,n}} > 1 - r_n$, that is, if $s_{pr,n}$ examples are drawn independently according to pr then with probability greater than $1 - r_n$ they are all in A_n. $pr(A_n) > 1 - r_n$, since $s_{pr,n} = s'_{pr,n+1} > 0$. Let f be any total function. Since S is dense, there is an $f' \in S$ which coincides with f on A_n. L approximates S, so with probability greater than $\frac{n+1}{n+2}$, L after having seen $s_{pr,n} = s'_{pr,n+1}$ examples coming from f', has put out its hypothesis e_{n+1}, and $\varphi_{e_{n+1}}$ differs from f' by less than $\frac{1}{n+2}$.

With probability greater than $1 - r_n$, f and f' coincide on the x-parts in the first $s_{pr,n}$ examples seen by L, hence with probability greater than $\frac{n+1}{n+2} - r_n = \frac{n}{n+1}$, M has produced the $n+1$'st hypothesis e_{n+1} of L as its n'th hypothesis, and $D_{pr}(\varphi_{e_{n+1}}, f') < \frac{1}{n+2}$. As $D_{pr}(f, f') \leq 1 - pr(A_n) < r_n$, we conclude that $D_{pr}(f, \varphi_{e_{n+1}}) < \frac{1}{n+1}$ with probability greater than $\frac{n}{n+1}$.

It is immediate from the construction that L's (possible) property to combine Ex-learning S (BC-learning S) with approximation extends to M's respective property of combining with universal approximation. $\square$

Our next result shows that for the "relatively weak" BC-learning, combining with universal approximation is always possible.

Theorem 4.3 (BC-Learning Combined). *Every BC-learnable class of total functions possesses a learner which combines BC-learning with universal approximation.*

Proof. From a BC-learner L of S, a new one M is constructed by interweaving L with our Universal Approximator U. Whenever U produces its hypothesis u_n as an index of a finite partial function σ_n, M patches σ_n with the last hypothesis of L known at that time. More particularly, given the infinite sequence $(x_1, y_1), (x_2, y_2), \ldots$, for any n let m_n be such that $U((x_1, y_1), \ldots, (x_{m_n}, y_{m_n})) = u_n$, and let a_n be L's last hypothesis at that time:

$a_n = L((x_1, y_1), \ldots, (x_l, y_l))$,
where l is the greatest $k \leq m_n$ with $L((x_1, y_1), \ldots, (x_k, y_k)) \downarrow \in \mathbb{N}$.

Let e_n satisfy

$$\varphi_{e_n}(x) = \begin{cases} \sigma_n(x) & \text{if } x \in dom(\sigma_n) \\ \varphi_{a_n}(x) & \text{otherwise} \end{cases}$$

and define M by

$$M((x_1, y_1), \ldots, (x_m, y_m)) = \begin{cases} e_n & \text{if } m = m_n \\ ? & \text{otherwise.} \end{cases}$$

All hypotheses e_n of M are such that φ_{e_n} is an extension of $\varphi_{u_n} = \sigma_n$, and they are put out at the same times as the respective u_n. Thus for any total f whatever, since U approximates f so does M.

Assume $f \in S$ and pr to be a fair probability distribution. Draw $(x_1, f(x_1))$, $(x_2, f(x_2)), \ldots$ according to pr. With probability 1, this sequence is valid and U outputs on it infinitely many hypotheses σ_n. Then, also M outputs infinitely many hypotheses. Since the function f extends every σ_n and L BC-learns f, M never makes errors caused by U and BC-learns f with probability 1, too. $\square$

Almost the same construction can be applied for Num, the classes which can be learned by enumeration. Recall from Example 2.3 that S is in Num iff S is a subclass of a recursively enumerable class of total recursive functions.

Theorem 4.4 (Enumeration Learning Combined). *Every class of total recursive functions that can be learned by enumeration possesses a learner which combines learning by enumeration with universal approximation.*

Proof. We can without loss of generality assume that $S = \{g_0, g_1, \ldots\}$ for a total recursive two-place function g, and that S is dense. From an enumeration learner L and our universal approximator U, M is defined as in the proof of Theorem 4.3 with the only difference that now e_n is such that

$$\varphi_{e_n} = g_k \text{ for the least } k \text{ satisfying } g_k(x) = \sigma_n(x) \text{ for all } x \in dom(\sigma_n).$$

M is a universal approximator since by the density of S each σ_n has a total extension g_k in S, and learns by enumeration every $f \in S$ because $f = g_k$ for some k. □

Let us now turn to Ex-learning, as the really interesting type of inductive convergence. The main reason for the possibility to combine BC-learning with approximation was that gradually improving the BC-hypotheses did not interfere with semantical convergence. But in the case of Ex-learning, syntactic convergence is required. As a default patching causes syntactic divergence, the Ex-learner must evaluate the current hypothesis with regard to local correctness in order to decide whether to update or not. This amounts to some kind of consistency.

Definition 4.5 (Weakly Consistent). L is **weakly consistent** iff for every x there is a number $b_x \geq x$ such that for all unambiguous sequences $((x_1, y_1), \ldots, (x_m, y_m))$ with $\{0, \ldots, b_x\} \subseteq \{x_1, \ldots, x_m\}$, $L((x_1, y_1), \ldots, (x_m, y_m)) \downarrow= e$ for some e such that $\varphi_e(x) = y_k$ for any k with $x_k = x$.

Note that one can assume without loss of generality, that a weakly consistent learner is total and never outputs ?; the above definition gives this already implicitly for those input sequences $((x_1, y_1), \ldots, (x_m, y_m))$ where $\{0, \ldots, b_0\} \subseteq \{x_1, \ldots, x_m\}$ and one could “hardwire” a default index for the remaining input sequences.

Our notation fits well with that from the literature. In fact, to be a *globally consistent* learner as defined by Jain, Osherson, Royer and Sharma [4, Definition 5.61 (d)] is equivalent to the special case $b_x = x$ in our definition above.

As a weakly consistent learner eventually withdraws every incorrect hypothesis, the learner either learns or outputs infinitely many different hypotheses. Minicozzi [6] has called learners with this latter property *reliable*. (Here we only consider reliable learning in the sense that the learner diverges on every total function it does not learn – this is different from other notions of reliability which do not postulate divergence on non-recursive functions.)

Our notion of weakly consistent learning is strictly in between globally consistent and reliable learning. For example, the class

$$\{f : (\exists e)[\varphi_e = f \wedge (\forall x)[e \text{ computes } f(x) \text{ in at most } f(f(x)) \text{ steps}]]\}$$

is reliable, but not weakly consistently Ex-learnable. The class

$$\{f : (\exists e)[\varphi_e = f \wedge (\forall x)[e \text{ computes } f(x) \text{ in at most } f(x+1) \text{ steps}]]\}$$

is weakly consistently but not globally consistently Ex-learnable.

Theorem 4.6 (Combining Ex-Learning with Approximation). *For a class S of total functions, the following are equivalent properties.*

(1) *There is a learner which combines Ex-learning of S with universal approximation.*

(2) *S is weakly consistently Ex-learnable.*

Proof. (1) $\Rightarrow$ (2). Let L be a learner which combines Ex-learning of S with universal approximation and fix the probability pr as $pr(\{x\}) = \frac{1}{x+1} - \frac{1}{x+2}$. Construct the weakly consistent learner M by induction. Let $M(\lambda) = 0$. For $m \geq 1$ and unambiguous input $((x_1, y_1), \ldots, (x_m, y_m))$, let $e' = M((x_1, y_1), \ldots, (x_{m-1}, y_{m-1}))$. Compute $e = M((x_1, y_1), \ldots, (x_m, y_m))$ as follows.

1: If $m \geq 3$, compute the probabilities of the following events $E_{k,n}$ where $3 \leq n \leq k \leq m$ and $E_{k,n}$ is the set of sequences $(z_1, \ldots, z_k)$ such that
- $z_1, \ldots, z_k \in \{x_1, \ldots, x_m\}$ and
- if L has seen $z_1, \ldots, z_k$ together with the corresponding y-values from the data $((x_1, y_1), \ldots, (x_m, y_m))$, then L has already output its n-th hypothesis e_n.

2: Choose the largest n such that either $n = 2$ or there is a k with $3 \leq n \leq k \leq m$ and $E_{k,n}$ has a probability of at least $\frac{n-1}{n+1}$.

3: Let e be as follows.
- If $m > 2$, $n > 2$ and the probability of the event $(E_{k,n} \wedge e' = e_n)$ is at least $\frac{2}{n+1}$ then let $e = e'$;
- else if $m > 2$, $n > 2$ and there are e'' such that the probability of the event $(E_{k,n} \wedge e'' = e_n)$ is at least $2^{-2-e''} + \frac{2}{n+1}$ then let e be the minimal such e'';
- otherwise let e be a program which is correct on all inputs seen so far.

Output e.

To see that the algorithm terminates, note that one has only to deal with the behaviour of L on sequences $z_1, \ldots, z_l$ together with the corresponding y-values where $l \leq m$ and $z_1, \ldots, z_l \in \{x_1, \ldots, x_m\}$. Hence the e'' considered in case 2 of step 3 must all occur in the output of L fed with one of these finitely many sequences.

Now take b_x to be the first number such that $b_x \geq x + s_{pr,(x+1)(x+2)+1}$ and

$$\left(\frac{b_x + 1}{b_x + 2}\right)^{s_{pr,(x+1)(x+2)+1}} > \frac{(x+1)(x+2)+1}{(x+1)(x+2)+2}.$$

Because of $\frac{b_x+1}{b_x+2} = pr(\{0, 1, \ldots, b_x\})$, the second condition ensures that if one draws $s_{pr,(x+1)(x+2)+1}$ many numbers according to pr, then all of these numbers are below b_x with probability greater than $\frac{(x+1)(x+2)+1}{(x+1)(x+2)+2}$. By definition of $s_{pr,(x+1)(x+2)+1}$, the probability that L has not put out $e_{(x+1)(x+2)+1}$ on the drawn data is at most $\frac{1}{(x+1)(x+2)+1}$. Thus $E_{s_{pr,(x+1)(x+2)+1},(x+1)(x+2)+1}$ has probability at least $\frac{(x+1)(x+2)}{(x+1)(x+2)+2}$.

Let the unambiguous input $((x_1, y_1), \ldots, (x_m, y_m))$ and (x, y) be given such that $\{0, \ldots, b_x\} \subseteq \{x_1, \ldots, x_m\}$ and $(x, y) = (x_l, y_l)$ for some l, and $e = M((x_1, y_1), \ldots, (x_m, y_m))$. By our remark on b_x, n in step 2 is at least $(x+1)(x+2)+1$. If, in step 3, e is selected according to one of the first two cases, then $e_n = e$ with probability at least $\frac{2}{(x+1)(x+2)+2}$. By definition of $s_{pr,(x+1)(x+2)+1}$ and as $pr(\{x\}) = \frac{1}{(x+1)(x+2)} > \frac{1}{n+1}$, it follows that $\varphi_e(x) \downarrow= y$. Otherwise e is selected according to the third case and correct at x as $b_x \geq x$. So M satisfies the weak consistency requirement.

Assume now that $f \in S$ and that $(x_1, f(x_1)), (x_2, f(x_2)), \ldots$ is a valid description for f. As M is weakly consistent, it is sufficient to show that M converges at all.

For every n' there is a value b such that whenever $\{0, \ldots, b\} \subseteq \{x_1, \ldots, x_m\}$ then the parameter n in step 2 is at least n'. So n goes to infinity when m goes to infinity. For every i, let r_i be the probability such that L converges on f to i (r_i is of course 0 if $\varphi_i \neq f$). As $r_0 + r_1 + \ldots = 1$, there is an i such that $r_i > 2^{-i-2}$. Then, for almost all m, the corresponding n, k from step 2 of the algorithm are such that the event $(E_{k,n} \wedge e_n = i)$ has probability greater than $\frac{2}{n+1} + 2^{-i-2}$. So, for almost all times where M changes its hypothesis, it outputs some $j \leq i$ according to the second case in step 3. Assume now by way of contradiction that M outputs some $j \leq i$ infinitely often but does not converge to j. There are infinitely many m where $e' \neq j$ but $e = j$ and e is selected according to the second case of step 3. For these m, the probability of the event $(E_{n,k} \wedge e_n = j)$ is at least 2^{-2-j}. There are also infinitely many m where $e' = j$ and $e \neq j$. Then the condition of the first case of step 3 does not hold and the event $(E_{n,k} \wedge e_n = j)$ has probability below $\frac{3}{n+2}$. Thus the measure of all sequences where L takes j infinitely often and abandons j infinitely often is at least 2^{-j-2}, a contradiction to L converging on f with probability 1. So M either outputs j only finitely often or outputs j almost always. M converges on f and, by the weak consistency of M, M learns f.

(2) ⇒ (1). Let L be a weakly consistent Ex-learner for S which without loss of generality is total and never outputs *?*. Let U be any universal approximator whose hypotheses $u_0, u_1, \ldots$ are always indices of finite partial functions (as was the case for the one constructed in Theorem 3.4). Recall that $\varphi_{e,m}(x)$ is defined to be $\varphi_e(x)$ if program e on argument x halts within m steps, and $\varphi_{e,m}(x) \uparrow$ otherwise. We define the total learner M on unambiguous inputs $((x_1, y_1), \ldots, (x_m, y_m))$ inductively as follows (where $M(\lambda) = ?$).

1: Let $n = |\{k < m : M((x_1, y_1), \ldots, (x_k, y_k)) \in \mathbb{N}\}|$ be the number of previous hypotheses of M. Compute $c = L((x_1, y_1), \ldots, (x_m, y_m))$. Check whether u_n is known, that is, whether $u_n = U((x_1, y_1), \ldots, (x_k, y_k))$ for some $k \leq m$.
2: Let $M((x_1, y_1), \ldots, (x_m, y_m))$ be as follows.
   - if u_n is known and $c \geq n$ then $M((x_1, y_1), \ldots, (x_m, y_m)) = u_n$;
   - else if u_n is known and $c < n$ and $\varphi_{c,m}(x) = \varphi_{u_n}(x)$ for all $x \in dom(\varphi_{u_n})$ then $M((x_1, y_1), \ldots, (x_m, y_m)) = c$;
   - otherwise $M((x_1, y_1), \ldots, (x_m, y_m)) = ?$.

Note that in the above definition $u_n = U((x_1, y_1), \ldots, (x_k, y_k))$, the domain of φ_{u_n} is $\{x_1, \ldots, x_k\}$. Thus $dom(\varphi_{u_n})$ can be computed as an explicit list by M.

Assume $f \in S$ and draw $(x_1, f(x_1)), (x_2, f(x_2)), \ldots$ according to pr. With probability 1, this sequence is valid and U outputs on it infinitely many hypotheses. As L is weakly consistent, each wrong hypothesis is eventually abandoned: Any current hypothesis c of L, for the m and n concerned, satisfies $(\forall x \in dom(\varphi_{u_n}))[\varphi_{c,m}(x) \downarrow = \varphi_{u_n}(x)]$ or will at some time be given up for ever. L Ex-learns f. So there is a hypothesis c and a number s of examples such that for all $m \geq s$,

- $L((x_1, f(x_1)), \ldots, (x_m, f(x_m))) = c$;
- $n > c$ for the number n of previous hypotheses of M.

It follows that from then on, M outputs either c or $?$. As M can for every n eventually verify that φ_c extends φ_{u_n}, M outputs c infinitely often and Ex-learns f. In particular, for any fair distribution pr to produce the examples, M Ex-learns S with probability 1.

It was not necessary in the above definition of M to fix the universal approximator U that we used, but in order to show that M is itself a universal approximator we shall now choose a particular such U. It is obtained by adapting the Basic Approximation Algorithm B in 3.4, i.e., replace it by a slightly sharper version B'. For pr, n define the following numbers $s'_{pr,n}$, $a_{pr,n}$, $b_{pr,n}$. $s'_{pr,n}$ is as $s_{pr,n}$ in Proposition 3.2 with the only exception that $s'_{pr,n}$ is now the number of examples needed to guarantee that B' has output its $X_{pr,n}$ with probability greater than $\frac{6n+5}{6n+6}$. $a_{pr,n}$ is the least z such that a drawn sequence $x_1, \ldots, x_{s'_{pr,n}}$ satisfies with probability at least $\frac{6n+5}{6n+6}$ that $x_j \leq z$ for $j = 1, \ldots, s'_{pr,n}$. $b_{pr,n} = \max\{x + b_x : x \leq a_{pr,n}\}$. Note that $b_{pr,n} \geq a_{pr,n}$. Now, let $s_{pr,n}$ be the first m such that the following holds with probability at least $\frac{6n+5}{6n+6}$.

- $m > s'_{pr,n}$,
- $\{0, \ldots, b_{pr,n}\} \subseteq \{x_1, \ldots, x_m\}$,
- $(\forall c < n)(\forall x \leq a_{pr,n})[\varphi_c(x)\downarrow \Rightarrow \varphi_{c,m}(x)\downarrow]$.

Then after $s_{pr,n}$ examples seen we have the following.

(a) With probability at least $\frac{6n+5}{6n+6}$, U has put out an index of a finite partial function σ_n.

(b) With probability at least $\frac{6n+5}{6n+6}$, $dom(\sigma_n) \subseteq \{x : x \leq a_{pr,n}\}$.

(c) With probability at least $\frac{6n+5}{6n+6}$, all of L's later hypotheses c will satisfy

$$(\forall x \leq a_{pr,n})[\varphi_c(x)\downarrow = f(x)].$$

(d) $(\forall c < n)(\forall x \leq a_{pr,n})[\varphi_c(x)\downarrow \Rightarrow \varphi_{c,s_{pr,n}}(x)\downarrow]$.

It follows that with probability at least $\frac{2n+1}{2n+2}$, (a), (b), (c), (d) are all satisfied after the occurrence of $s_{pr,n}$ examples. By the definition of M, M has then produced a hypothesis which extends the hypothesis σ_n of U. We can now argue as in the proof of Proposition 3.2 and Theorem 3.4 to conclude that with probability greater than $\frac{n}{n+1}$, M after $s_{pr,n}$ examples has output its n'th hypothesis e_n, and this satisfies $D_{pr}(f, \varphi_{e_n}) < \frac{1}{n+1}$. $\square$

It follows immediately from the definition of weak consistency that every dense class which can be learned by enumeration is weakly consistently Ex-learnable (with $b_x = x$). As density can be assumed w.l.o.g. in the case of learning by enumeration, we obtain Theorem 4.4 again as a corollary to Theorem 4.6. Weak consistency is in fact a quite restrictive property, which can be seen by considering $\{0,1\}$-valued functions.

Corollary 4.7 ($\{0,1\}$-Valued Functions). *For a class S of total recursive $\{0,1\}$-valued functions, the following are equivalent properties.*

(1) *There is a learner which combines Ex-learning of S with universal approximation;*
(2) *S is weakly consistently Ex-learnable;*
(3) *S is enumeration learnable.*

Proof. It is well-known [13] that every reliably learnable class of $\{0,1\}$-valued functions is learnable by enumeration. $\square$

Remember the class of self-describing functions, which was Ex-learnable in quite a trivial way but not learnable by enumeration. If we weaken universal approximation to just approximation on S then Ex-learning that class can in fact be combined with approximation.

Example 4.8 (Self-Describing Functions Approximated). *The class S of self-describing functions has a learner which combines Ex-learning of S with approximation.*

Proof. L is rather simple-minded. It waits until $(0, f(0))$ has occurred in a sequence of examples and from then on continues to output $f(0)$. For any fair probability pr and $n \in \mathbb{N}$, there is an $s_{pr,n}$ such that $(0, f(0))$ will have appeared with probability greater than $\frac{n}{n+1}$ for all total functions f. If $f \in S$ then $f(0)$ is an index of f, hence f is certainly approximated (even "exactly hit") from then on. $\square$

The self-describing functions do not form a dense class, and in fact, density would immediately invalidate the above method:

Corollary 4.9 (Dense Classes). *For a dense class S of recursive functions, the notions of*

- *Ex-learnability combined with approximation*

and of

- *Ex-learnability combined with universal approximation*

coincide.

Proof. Immediate by Corollary 4.2 and Theorem 4.6. $\square$

What about an Ex-learnable class, none of whose Ex-learners can be combined with approximation? One easily proves that the class of self-describing functions is not reliably Ex-learnable, hence Ex-learning this class cannot be combined with universal approximation. On the following dense Ex-learnable class S, then, Ex-learning cannot be combined with approximation; S may serve as a witness to separate Ex-learnability from Ex-learnability combined with approximation.

Example 4.10 (Ex-learning Not Combinable). *The class S containing all functions f such that*

- *either there is a self-describing g satisfying $(\forall x)[f(x) = g(x) + 1]$,*
- *or $f(x) = 0$ for almost all x*

is Ex-learnable, but no Ex-learner of it can be combined with approximation.

We omit the easy proof (use Theorem 4.6 and Corollary 4.9).

Remark 4.11 (Weakly Consistent on S). One can restrict the notion of weak consistence to a class S of functions by adapting Definition 4.5 as follows. A learner L is **weakly consistent on** S iff there is for every x a number $b_x \geq x$ such that for all $f \in S$ and all sequences $((x_1, f(x_1)), \ldots, (x_m, f(x_m)))$ satisfying $\{0, \ldots, b_x\} \subseteq \{x_1, \ldots, x_m\}$, $L((x_1, f(x_1)), \ldots, (x_m, f(x_m)))$ outputs a hypothesis e such that $\varphi_e(x) \downarrow= f(x)$. Now one can generalize Theorem 4.6 by showing that the following properties are equivalent for any class S.

(1) Some learner of S combines Ex-learning of S with approximation.
(2) There is an Ex-learner for S which is weakly consistent on S.

Note that in both cases the learner can be partial; but the learner is undefined only if the input is not derived from any function in S.

5 Distribution-Free Approximation

Thus far, approximation was such that bounds on the numbers of drawn examples, to ensure a certain quality of approximation, were allowed to depend on the underlying distribution. We will now deal with the case of distribution-independent such bounds. If such a demand is combined with the uniformity w.r.t. the target function f (as in Definition 3.3), one arrives at a rather extreme case. In order to be not too restrictive from the beginning, we first consider a notion, so to say, "orthogonal" to approximation from Definition 3.3: Quality bounds are independent from pr, but may depend on the target function f.

Definition 5.1 (Distribution-Free Approximation). A learner L **approximates** a total function f **in a distribution-free way** iff for all $n \in \mathbb{N}$ there exists a bound $s_{f,n}$ such that for all distributions pr on $\mathbb{N}$ if examples are read according to pr_f the following holds with probability greater than $\frac{n}{n+1}$: L outputs its n'th hypothesis e_n after at most $s_{f,n}$ examples read, and $D_{pr}(\varphi_{e_n}, f) < \frac{1}{n+1}$. Let T be a type of (inductive) learning and S a class of total functions. L **combines T-Learning of S with distribution-free approximation** iff L T-learns S with probability 1 and approximates every $f \in S$ in a distribution-free way. L **combines T-learning of S with uniform distribution-free approximation** iff L combines T-learning of S with distribution-free approximation in such a way that the bounds $s_{f,n}$ belonging to the $f \in S$ can be chosen independently from f, i.e., as constants s_n.

Definition 5.2 (Consistent). L is **consistent on** S iff for all $f \in S$ and sequences $((x_1, f(x_1)), \ldots, (x_m, f(x_m)))$, $L((x_1, f(x_1)), \ldots, (x_m, f(x_m)))$ is defined and outputs a hypothesis e such that $\varphi_e(x_k) = f(x_k)$ for $k = 1, \ldots, m$. L is **consistent** iff L is consistent on the class of all total functions, i.e., iff for all unambiguous sequences $((x_1, y_1), \ldots, (x_m, y_m))$, $L((x_1, y_1), \ldots, (x_m, y_m))$ outputs a hypothesis e with $\varphi_e(x_k) = y_k$ for $k = 1, \ldots, m$.

Remark 5.3 (Consistency and Order). While for many learning criteria, one can normalize the sequences under investigation to the form $(0, f(0))$, $(1, f(1)), \ldots$ without changing the concept, this does not work for consistency. Although most publications in inductive inference follow the consistency notion based on normalized input, we have picked for our work the more restrictive version above which deals with all possible input sequences. Recall that global consistency on normalized input is equivalent to our notion of weakly consistent learning with $b_x = x$ for all x.

Theorem 5.4 (Ex-Learning Combined with Distribution-Free Approximation). *If S has a learner that combines Ex-learning with distribution-free approximation then there is an Ex-learner of S which is consistent on S.*

Proof. Let L be a learner for S that combines Ex-learning with distribution-free approximation. Without loss of generality, we describe our desired learner M with regard to some given infinite sequence $(x_1, y_1), (x_2, y_1), \ldots$ – if for some $f \in S$ all y_k are $f(x_k)$ then a hypothesis $e_m = M((x_1, y_1), \ldots, (x_m, y_m))$ will be output for every m, otherwise $M((x_1, y_1), \ldots, (x_m, y_m))$ may become undefined. The hypotheses e_m are computed by induction on m, along with numbers s_m and rationals ϵ_m, such that $s_0 < s_1 < \ldots$ and $\epsilon_0 > \epsilon_1 > \ldots > 0$. s_m stands for the number of randomly drawn examples in a simulation of L and the ϵ_i serve to gradually define a distribution pr together with approximations pr_j to pr by putting for $i, j \in \{1, \ldots, m\}$,

$$pr(x_i) = \epsilon_{i-1} - \epsilon_i;$$

$$pr_j(x_i) = \begin{cases} \epsilon_{i-1} - \epsilon_i & \text{if } i < j; \\ \epsilon_{i-1} & \text{if } i = j; \\ 0 & \text{if } i > j. \end{cases}$$

Now M proceeds as follows. $s_0 = 0$, $\epsilon_0 = 1$, $e_0 = M(\lambda) = 0$. For $m > 0$, assume that s_i, ϵ_i and e_i have already been computed for $i = 0, \ldots, m-1$ such that $s_0 < s_1 < \ldots < s_{m-1}$, $\epsilon_0 > \epsilon_1 > \ldots > \epsilon_{m-1} > 0$ and $e_i = M((x_1, y_1), \ldots, (x_i, y_i))$. To compute s_m, ϵ_m, e_m, do the following.

1: Find the first $s_m > s_{m-1}$ such that there is n satisfying the following conditions where the underlying probability is pr_m (and not the at x_m still unknown pr).
   - $\frac{2}{\epsilon_{m-1}} < n < s_m$;
   - the sum p_m of the probabilities $p_{m,e}$ is at least $\frac{n}{n+1}$ where $p_{m,e}$ is the probability that L outputs at least n hypotheses and the n'th hypothesis is e;
   - if $p_{m,e} > \frac{2}{n+1}$ then $\varphi_e(x) = y$ within s_m computation steps for all (x, y) in $\{(x_1, y_1), \ldots, (x_m, y_m)\}$.

   Fix any such n.

2: Among the following cases, choose the first one applying and output the corresponding hypothesis e_m.
   - Let $e_m = e_{m-1}$ if $p_{m,e_{m-1}} > \frac{2}{n+1}$;
   - Let e_m be the minimal number e such that $p_{m,e} > \frac{2}{n+1} + 2^{-e-2}$ if such an e exists;
   - Otherwise let e_m be a hypothesis consistent with all data seen so far.

3: Choose $\epsilon_m = \frac{1}{k}$ for the first k with $(\frac{k-1}{k})^{s_m} > \frac{2n+1}{2n+2}$.

This algorithm is verified in a similar way as the one in part $(1) \Rightarrow (2)$ of the proof of Theorem 4.6:

This learner M is computable as for every s_m one has only to consider m^{s_m} possible input sequences of length s_m and so rationals representing the values $p_{m,e}$ can be computed explicitly. In particular only finitely many $p_{m,e}$ are greater than 0 and the e and the corresponding $p_{m,e}$ can be listed explicitly. Nevertheless, M needs L to be defined on the simulated inputs and s_m to exist – which is only guaranteed if the data is from a function $f \in S$. Furthermore, M is consistent on S: Either the index e_m satisfies $p_{m,e_m} > \frac{2}{n+1}$ and consistency follows from the third search condition in the second step. Or e_m is picked explicitly to be correct on all data seen so far.

Assume now $f \in S$. Then L learns f. As M is consistent on S, it is sufficient to show that M converges at all. Note that n goes to infinity when m goes to infinity.

For every e, L converges to e with some probability r_e. Furthermore, the values $p_{m,e}$ converge to r_e for $m \to \infty$. If $r_e = 0$ and M does not converge to e, then e only finitely often qualifies in case 2 of step 3 and $e \neq e_m$ for almost all m. If $r_e > 0$ and $e = e_m$ for infinitely many m, then M converges to e as $p_{m,e} > \frac{2}{n+1}$ for almost all m and the corresponding n. So, for every e, it holds either finitely often or almost always that $e = e_m$.

As $r_0 + r_1 + \ldots = 1$, there is an e with $r_e > 2^{-2-e}$. It follows that for almost all m, either $e_m = e_{m-1}$ or $e_m \leq e$ as some $e'' \leq e$ qualifies in case 2 of step 3. So the sequence of the e_m is bounded and thus converges. $\square$

For dense classes, the converse of Theorem 5.4 is also true. Observe that for any learner L, consistency on a dense class S is equivalent to consistency. We have the following characterization

Theorem 5.5 (Distribution-Free Approximation on Dense Classes). *The following properties of a dense class S of total recursive functions are equivalent.*

- **(a)** *There is a learner which combines Ex-learning of S with distribution-free approximation;*
- **(b)** *There is a consistent Ex-learner of S;*
- **(c)** *There is a numbering ψ of partial recursive functions such that $S \subseteq \{\psi_e : e \in \mathbb{N}\}$ and the set $\{(e, x, y) : \psi_e(x) \downarrow = y\}$ is recursive.*

Proof. (a) $\Rightarrow$ (b): Immediate by Theorem 5.4, as S is dense.

(b) $\Rightarrow$ (c): Let the consistent Ex-learner L of S be given. Observe that L must be total. The numbering ψ is based on a one-to-one recursive enumeration σ_e, $e = 0, 1, \ldots$, of all unambiguous finite sequences of pairs of natural numbers. If $\sigma_e = ((x_1, y_1), \ldots, (x_m, y_m))$ then define $\psi_e(x) \downarrow = y$ iff

- either $(x, y) \in \{(x_1, y_1), \ldots, (x_m, y_m)\}$
- or $x \notin \{x_1, \ldots, x_m\} \wedge L((x_1, y_1), \ldots, (x_m, y_m)) = L((x_1, y_1), \ldots, (x_m, y_m), (x, y))$.

By construction, $\{(e, x, y) : \psi_e(x) \downarrow = y\}$ is recursive, and every f, which is Ex-learned by L, is in $\{\psi_e : e \in \mathbb{N}\}$.

(c) $\Rightarrow$ (a): Let ψ be given. The learner M defined below combines a generalized principle of learning by enumeration with a quality test based on the Chernoff bound. In order to simplify notation we let M output ψ-indices instead of φ-indices. M's computing of its hypothesis e_n (as a random variable) is described by induction on n.

$e_0 = M(\lambda) = 0$. For $n \geq 1$, assume that e_{n-1} has already been defined, and let $e' = e_{n-1}$. To compute e_n, M proceeds as follows.

1: Read $cher(\frac{2^{-e'-1}}{n+1}, \frac{1}{n+1})$ new examples (producing *?* as long as this has not been finished);
2: If $\psi_{e'}(x_k) = y_k$ for all examples (x_k, y_k) seen so far then output $e_n = e'$ as hypothesis, otherwise output *?*, update $e' = e' + 1$, and go to 1.

Obviously, M is consistent and Ex-learns S. In order to show that M approximates every $f \in S$ in a distribution-free way, let e be the minimal ψ-index of f. Then for examples w.r.t. any pr_f, e' in the above algorithm will never exceed e, hence

$$\sum_{n'=0}^{n} \sum_{e'=0}^{e} cher\left(\frac{2^{-e'-1}}{n'+1}, \frac{1}{n'+1}\right)$$

is an upper bound for the examples read until $e_0, e_1, \ldots, e_n$ have all been output. In order to estimate the probability that $D_{pr}(\psi_{e_n}, f) \geq \frac{1}{n+1}$, observe that this can only happen if $e_n < e$ and for some e' (that one with $e_n = e'$), $D_{pr}(\psi_{e'}, f) \geq \frac{1}{n+1}$, which happens with probability at most $\frac{2^{-e'-1}}{n+1}$. Hence the overall probability that $D_{pr}(\psi_{e_n}, f) \geq \frac{1}{n+1}$ is at most $\sum_{e' \leq e} \frac{2^{-e'-1}}{n+1} < \frac{1}{n+1}$. $\square$

Theorem 5.4 cannot be transferred to non-dense classes. For example, consider the following class of functions satisfying a modified version of self-description.

$$S = \{f : (\forall x)(\exists e, z)[f(x) = \langle e, z \rangle \wedge f = \varphi_e]\},$$

where $\langle \ldots \rangle$ is some standard computable pairing function. S is obviously an Ex-learner which is consistent on S, but an easy diagonal argument shows that there is no ψ such that $\{(e, x, y) : \psi_e(x) \downarrow = y\}$ is recursive and $\{\psi_e : e \in \mathbb{N}\}$ contains S.

Let us at last turn to uniform distribution-free approximation, i.e., the extremely restrictive property that the bound on the number of examples to guarantee a desired level of approximation can be chosen independently both from pr and f. Apart from complexity requirements, this is the original version of a learner to be PAC (probably approximately correct) [10], but the notion of PAC-learnability has later been extended to more general situations [12]. To avoid misunderstanding, we therefore keep to our nomenclature from Definition 5.1.

The main results concerning uniform distribution-free approximation are connected to the concept of the *VC-dimension* of a class of $\{0, 1\}$-valued functions (see Definition 5.6 below), or *P-dimension* in the general case. As for $\{0, 1\}$-valued functions, uniform distribution-free learnability is equivalent to having finite VC-dimension, and for general functions, finiteness of P-dimension is sufficient but not necessary [11,12]. These results are obtained in a quite general setting, so that they hold true in our special case of functions from $\mathbb{N}$ to $\mathbb{N}$, too. Still, this special situation might possibly allow for a simpler proof. Below, we give a relatively short proof that any uniformly and distribution-freely approximable class of $\{0, 1\}$-valued functions must have finite VC-dimension.

Definition 5.6 (VC-Dimension). The **VC-dimension** (Vapnik-Chervonenkis dimension) of a class T of total functions from $\mathbb{N}$ to $\{0, 1\}$ is the greatest k satisfying

- there is a set X of cardinality k such that every function $\psi : X \to \{0,1\}$ is extended by some $f \in T$

if only finitely many k have this property, and is infinite otherwise.

Fact 5.7 (Uniform Distribution-Free Approximation of $\{0,1\}$-valued Functions). *If a class T of total functions $\mathbb{N} \to \{0,1\}$ can be approximated in a uniform, distribution-free way then T has finite Vapnik-Chervonenkis dimension.*

Proof. Let L approximate T in a uniform and distribution-free way, and s_n the bounds as required by this property. Let k be any multiple of 8, and let $X \subseteq \mathbb{N}$ be such that $|X| = k$ and every $\psi : X \to \{0,1\}$ has an extension in T. Consider $F \subseteq T$ such that every $\psi : X \to \{0,1\}$ has a unique extension in F, hence $|F| = 2^k$.

One can choose $m = 8^{k/8}$ functions $f_1, \ldots, f_m \in F$ which pairwise have Hamming distance of at least $\frac{k}{8}$ on X: Given any $f \in F$, consider the set $G(f)$ of all $g \in F$ such that the Hamming distance of f, g on X is at most $k/8$. The cardinality of $G(f)$ is given by

$$|G(f)| = \sum_{h=0}^{k/8} \binom{k}{h}$$

where, by $h \leq k/8$,

$$\binom{k}{h} = \prod_{l=1}^{h} \frac{k+1-l}{l} \leq \prod_{l=1}^{h} \frac{8 \cdot (k/4+1-l)}{l} \leq 8^{k/8} \cdot \binom{k/4}{h}.$$

This inequality can be used to obtain

$$|G(f)| \leq 8^{k/8} \cdot \sum_{h=0}^{k/8} \binom{k/4}{h} \leq 8^{k/8} \cdot 2^{k/4} = 32^{k/8}.$$

Choose $f_1 \in F$ arbitrarily and, for $i = 2, 3, \ldots$, take f_i from the set $F - \bigcup_{j=1}^{i-1} G(f_j)$ until an i is reached where this set becomes empty. Since $|F| = 2^k = 256^{k/8}$ and $\bigcup_{j=1}^{i-1} G(f_j)$ contains at most $(i-1) \cdot 32^{k/8}$ many functions, at least $m = 8^{k/8}$ many functions $f_1, \ldots, f_m$ are found.

Now let pr_X be such that in any sequence $x_1, x_2, \ldots$ the first s_{16} examples are, with a probability greater than $\frac{16}{17}$, in X, and $pr_X(\{x\}) = pr_X(\{x'\})$ for all x, x' in X. By assumption, L after s_{16} examples has with a probability greater than $\frac{16}{17}$ identified any $f \in T$ up to an error of less than $\frac{1}{17}$. Hence with probability at least $\frac{15}{17}$, if s_{16} examples $(x_1, f(x_1)), \ldots, (x_{s_{16}}, f(x_{s_{16}}))$ are drawn, then $x_1, \ldots, x_{s_{16}} \in X$, L's 16'th hypothesis e_{16} has been put out and $D_{pr_X}(f, \varphi_{e_{16}}) \leq \frac{1}{17}$. For $f = f_i$, if $(x_1, f_i(x_1))$, $\ldots, (x_{s_{16}}, f_i(x_{s_{16}}))$ has this property, then this is not the case for f_j, $j \neq i$, because of $D_{pr_X}(f_i, f_j) \geq pr_X(X) \cdot \frac{1}{8} > \frac{2}{17}$. As all examples in the sequences have the form (x, y) with $y \in \{0,1\}$, there are at most $(2k)^{s_{16}}$ possible sequences of length s_{16}, so that k must satisfy $8^{k/8} \leq (2k)^{s_{16}}$. The first term is exponential in k and the second one polynomial. Thus the Vapnik-Chervonenkis dimension of T is finite. □

Having finite VC-dimension is an extremely restrictive property (apart from finite classes, an example of a "natural" class which has it is, for a given constant c, the class of all functions f which are 0 at all but at most c places). Our next example shows that even this (somehow) most restrictive kind of approximability does not, in general, imply inductive learnability, not even in the weak sense of BC-learning. Thus despite all of our "positive" results, an enormous gap between inductive learning and approximation remains. Below, $\langle \ldots \rangle$ is a recursive one-to-one mapping from the set $\mathbb{N}^+$ of all non-empty strings onto $\mathbb{N}$.

Example 5.8 (Finite VC-Dimension Does Not Imply BC-Learnability). *Let S be the class of all functions g_f, $f \in$ REC, where*

$$g_f(x) = \begin{cases} 1 & \text{if } x = \langle f(0), f(1), \ldots, f(k) \rangle \text{ for some } k; \\ 0 & \text{otherwise, that is, there is no such } k. \end{cases}$$

S has VC-dimension 1 but is not BC-learnable.

Proof. S has VC-dimension 1: Consider $\{x, y\}$ with $x \neq y$. We have to show that there is some $\sigma : \{x, y\} \to \{0, 1\}$ which is not extended by any $g \in S$ ($\{x, y\}$ is not "shattered by S"). $x = \langle a_0, \ldots, a_k \rangle$ for unique numbers $k, a_0, \ldots, a_k$. If $y = \langle a_0, \ldots, a_l \rangle$ for some $l < k$ then $g(x) = 1 \Rightarrow g(y) = 1$ for all $g \in S$; choose $\sigma(x) = 1$, $\sigma(y) = 0$. If $y = \langle a_0, \ldots, a_l \rangle$ for some $l > k$ and numbers $a_{k+1}, \ldots, a_l$ then $g(y) = 1 \Rightarrow g(x) = 1$ for all $g \in S$; choose $\sigma(x) = 0$, $\sigma(y) = 1$. If y is of none of these two forms, then $g(x) = 0 \vee g(y) = 0$ for all $g \in S$; choose $\sigma(x) = 1$, $\sigma(y) = 1$.

It is well known that REC is not BC-learnable [1,4,7,8]. An easy reduction shows that BC-learnability of S would simply imply BC-learnability of REC: Given $f \in$ REC, compute the arguments for g_f from those for f, apply the (assumed) BC-learner of S and translate produced hypotheses in such a way that e is a program for g_f iff its translation is a program for f. $\square$

Remark 5.9 (Questions to the Reader). Concerning the "learning as uniform approximation" paradigm of statistical learning theory, we have arrived at a somewhat strange situation. Over a countable domain such as $\mathbb{N}$, there are two possibilities.

Either one allows quality bounds (on the numbers of needed examples) to depend on the distribution. Then the whole class of all total functions from $\mathbb{N}$ to $\mathbb{N}$ becomes "learnable" (uniformly approximable). Even stronger, the learning algorithm itself does *not* need to depend on pr; only the "environment" it acts in influences convergence. Thus, *nothing specific of "learning" seems to be left at all*. Also, all mathematical tools needed are from standard statistics (the Chernoff bound).

Or, secondly, one requires those bounds to not depend on distributions. Then the whole concept of "learning", so to say, collapses the other way round: Extremely few classes remain learnable in this sense (i.e., uniformly approximable in a distribution-free way), in the case of $\{0, 1\}$-valued functions just those of finite VC-dimension.

There might be ways out of the dilemma. E.g., to consider additive but not σ-additive measures. Anyway, a lot of questions remain, and combining with inductive inference might become interesting anew.

References

1. Janis Bārzdiņš. Two theorems on the limiting synthesis of functions. In Latvian State University, editor, *Theory of Algorithms and Programs*, volume I, pages 82–88. 1974. In Russian.
2. John Case and Carl Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.
3. E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
4. Sanjay Jain, Daniel Osherson, James Royer, and Arun Sharma. *Systems That Learn*. The MIT Press, Cambridge, Massachusetts, 1999. revised edition of [8].
5. W. Menzel and F. Stephan. Topological aspects of numberings. *Mathematical Logic Quarterly*. to appear. See also: Interner Bericht 15, Universität Karlsruhe, Fakultät für Informatik, Karlsruhe, 1999.
6. Eliana Minicozzi. Some natural properties of strong-identification in inductive inference. *Theoretical Computer Science*, 2:345–360, 1976.
7. Piergiorgio Odifreddi. *Classical Recursion Theory*, volume I and II. North-Holland and Elsevier, Amsterdam, 1989 and 1999.
8. Daniel Osherson, Michael Stob, and Scott Weinstein. *Systems That Learn. An Introduction to Learning Theory for Cognitive and Computer Scientists*. Bradford – The MIT Press, Cambridge, Massachusetts, 1986.
9. Robert I. Soare. *Recursively Enumerable Sets and Degrees. A Study of Computable Functions and Computably Generated Sets*. Springer, Heidelberg, 1987.
10. Leslie G. Valiant. A theory of the learnable. *Communications of the Association for Computing Machinery*, 27(11):1134–1142, 1984.
11. Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.
12. Mathukumalli Vidyasagar. *A Theory of Learning and Generalization*. Springer, London, 1997.
13. Thomas Zeugmann. A-posteriori characterizations in inductive inference of recursive functions. *Journal of Information Processing and Cybernetics* (EIK), 19:559–594, 1983.